ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Title goes here

Master Thesis

Robin Bosshard, 16-915-399

October 16, 2023

Supervisors: Prof. Dr. Fernando Perez-Cruz, Dr. Eliza Harris, Lili Gasser (SDSC)
Dr. Kasia Arturi (Eawag)

Department of Computer Science, ETH Zürich

Contents

Chapter 1

Material and Methods

1.1 Toxicity Data

1.1.1 ToxCast invitroDB v4.1

The most recent release of the ToxCast's database[1], referred to as *invitroDBv4.1*, serves as a source of an extensive collection of HTS targeted bioactivity data. This database encompasses information on a total of 10 196 compounds, selectively tested across 1485 assay endpoints. The assays utilize a range of technologies to assess the impact of chemical compounds on a wide array of biological targets, including individual proteins and cellular processes such as mitochondrial health, developmental processes and nuclear receptor signaling.

This resource originated from the collaboration of two prominent institutions: the U.S. EPA through its ToxCast program and the National Institutes of Health (NIH) via the Tox21 initiative. Using data collected from multiple research laboratories (refer to Table A.1 in the Appendix), this relational database is accessible to the public and can be downloaded[2] by visiting the official ToxCast website.

1.1.2 tcpl v3.0

The *tcpl*[3] package provides a wide range of tools for efficiently managing HTS data. It enables reproducible concentration-response modeling and populates the `MYSQL` database, invitroDBv4.1. The multiple-concentration screening paradigm intends to pinpoint the activity of compounds, while also estimating their efficacy and potency.

In Section 1.2, we introduce *pytcpl* a `Python` re-implementation of the vital components that underpin the entire ToxCast pipeline. It should be noted that these components, as presented in the following, are applicable to both tcpl and pytcpl.

[1] released on September 21, 2023

[2] https://www.epa.gov/chemical-research/exploring-toxcast-data

[3] https://github.com/USEPA/CompTox-ToxCast-tcpl

1.1.3 Concentration-Response Series

Each compound-assay dataset involves the collection of the respective *concentration-response series (CRS)* denoted as CRS_{ij}, representing compound c_j in assay endpoint a_i. A CRS is represented as a set of concentration-response pairs:

$$CRS_{i,j} = \{(conc_{1_{i,j}}, resp_{1_{i,j}}), (conc_{2_{i,j}}, resp_{2_{i,j}}), \ldots, (conc_{n_{\text{datapoints}_{i,j}}}, resp_{n_{\text{datapoints}_{i,j}}})\}$$

where $n_{\text{datapoints}_{i,j}}$ varies based on the number of concentrations tested for compound c_j in the assay endpoint a_i. Figure 1.1 showcases a single CRS for some compound tested within an assay endpoint.

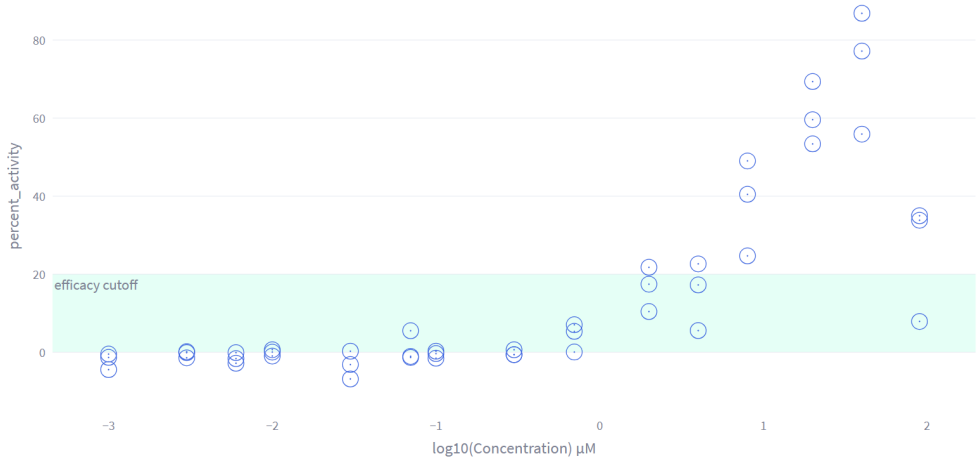


Figure 1.1: The concentration-response series (CRS) belongs to the compound *Diofenolan* (DTXSID2041884), tested in the assay endpoint emphTOX21_ERa_LUC_VM7_Agonist, identified by the assay endpoint ID $aeid = 788$. This particular series comprises a total of $k = 45$ concentration-response pairs and is structured into $n_{conc} = 15$ distinct concentration groups, with each group consisting of $n_{rep} = 3$ replicates.

In practice, concentrations are often subjected to multiple testing iterations, resulting in the distinct concentration groups with replicates. Table 1.1 presents the key quantities associated with an individual CRS when considering a specific assay endpoint a_i and compound c_i. Additionally, to visualize the variations in these metrics across the complete set of analyzed CRS in this thesis, please refer to Figure A.1 in the Appendix.

Concentration-response pairs, along with essential sample information such as well type and assay well-plate indices, can be retrieved by combining tables *mc0*, *mc1*, and *mc3* from invitroDBv4.1, which represent the raw data. Therby a special role have the control wells that typically contain untreated samples or samples with a known, non-toxic response. They are used as a baseline to normalize the treated samples and account for any background noise in the assay [1]. The concentrations are transformed to the logarithmic scale having the unit μM (micromolar), while the responses are control well-normalized to either *fold-induction* or *percent-of-control* activity:

Quantity	Description
$n_{\text{datapoints}_{i,j}}$	Total number of concentration-response pairs ($\|CRS\|$)
$n_{\text{groups}_{i,j}}$	Number of distinct concentrations tested
$n_{\text{replicates}_{i,j}}$	Number of replicates for each concentration group
$min_{\text{conc}_{i,j}}$	Lowest concentration tested
$max_{\text{conc}_{i,j}}$	Highest concentration tested

Table 1.1: Description of Parameters

1. **Fold Induction:** is a measure used to quantify how much, for instance, gene expression or protein activity has changed in response to a treatment compared to its baseline level from the control well set. For example, if a gene is expressed five times higher in a treated sample compared to the control, the fold induction would be 5.
2. **Percent of Control:** is another way to express the relative change in a activity due to a treatment.

1.1.4 tcplFit2

TcplFit2[4] is an extension to tcpl, focused on curve-fitting and hit-calling. The package also offers a flexible and robust fitting procedure, allowing for the use of different optimization algorithms and the incorporation of user-defined constraints. This sets it apart from other open-source concentration-response packages such as *drc* and *mixtox*, as it is explicitly designed for HTS concentration-response data.

1.1.5 Curve Fitting

All the curve fit models from tcplFit2, as outlined in Table 1.2, assume that the normalized observations in the CRS conform to a Student's t-distribution with 4 degrees of freedom [2]. The Student's t-distribution has heavier tails compared to the normal distribution, making it more robust to outlier and eliminates the necessity of removing potential outliers prior to the fitting process. The model fitting algorithm in tcplFit2 employs nonlinear *maximum likelihood estimation (MLE)* to determine the model parameters for all available models.

Consider $t(z, \nu)$ as the Student's t-distribution with ν degrees of freedom, where y_i represents the observed response for the i-th observation, and μ_i is the estimated response for the same observation. The calculation of z_i is as follows: $z_i = y_i - \mu_i \exp(\sigma)$, where σ is the scale term. Then the log-likelihood is: $\sum_{i=1}^{n}[\ln(t(z_i, 4)) - \sigma]$, where n is the number of observations.

The *Akaike Information Criterion (AIC)* is used as measure of goodness of fit, defined by the formula: $AIC = -2\log(L(\hat{\theta}, y)) + 2K$. The model with the lowest AIC value

[4] https://github.com/USEPA/CompTox-ToxCast-tcplFit2

Table 1.2: tcplfit2 Model Details

Model	Label	Equations[1]
Constant	cnst	$f(x) = 0$
Linear	poly1	$f(x) = ax$
Quadratic	poly2	$f(x) = a\left(\frac{x}{b} + \left(\frac{x}{b}\right)^2\right)$
Power	pow	$f(x) = ax^p$
Hill	hill	$f(x) = \frac{tp}{1+\left(\frac{ga}{x}\right)^p}$
Gain-Loss	gnls	$f(x) = \frac{tp}{(1+\left(\frac{ga}{x}\right)^p)(1+\left(\frac{x}{la}\right)^q)}$
Exponential 2	exp2	$f(x) = a\left(\exp\left(\frac{x}{b}\right) - 1\right)$
Exponential 3	exp3	$f(x) = a\left(\exp\left(\left(\frac{x}{b}\right)^p\right) - 1\right)$
Exponential 4	exp4	$f(x) = tp\left(1 - 2^{-\frac{x}{ga}}\right)$
Exponential 5	exp5	$f(x) = tp\left(1 - 2^{-\left(\frac{x}{ga}\right)^p}\right)$

[1] Parameters: a: x-scale, b: y-scale p: (gain) power, q: (loss) power, tp: top, ga: gain AC50, la: loss AC50

is chosen as the *winning* model. The winning model is then used to estimate the efficacy and potency of the compound. The potency estimates, also called *point-of-departure (POD)* estimates, are derived from the fitted curve, identifying certain *activity concentrations (AC)* at which the curve crosses certain response levels. In particular, the following POD estimates are computed:

- the AC at which the compound first reaches the estimated efficacy cutoff is denoted by *acc*
- the AC at which the compound first reaches 50% of its estimated maximum response is denoted by *ac50*
- the AC at which the compound first reaches its median maximum response is denoted by *actop*

Figure 1.2 illustrates the stated POD estimates. Notably, no POD estimates are computed when the compound is predicted inactive.

1.1.6 Hit Calling

The *continuous hitcall* is a measure of the probability that a compound is active, calculated based on the product of the probability values [1]:

i. that at least one median response is greater than the efficacy cutoff, computed

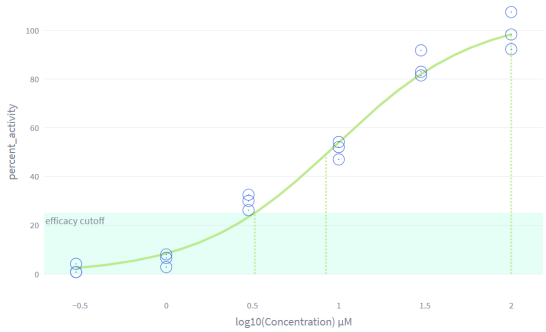


Figure 1.2: *Point-of-departure (POD)* estimates for the chemical *Picoxystrobin* (DTXSID9047542) in the assay endpoint *OT_FXR_FXRSRC1_0480* (aeid=753). The efficacy cutoff is set to 25 percent-of-control activity. The winning fit model is the Hill function and acc, ac50 and actop, have been estimated to be 3.3e+00, 8.4e+00 and 1.0e+02 $\log_{10} \mu M$, respectively.

by using the error parameter from the model fit and Student t-distribution to calculate the odds of at least one response exceeding the efficacy cutoff;

ii. that the top of the winning fitted curve is above the cutoff which is the likelihood ratio of the one-sided probability of the efficacy cutoff being exceeded;

iii. that the winning AIC value is less than that of the constant model:

$$\frac{e^{-\frac{1}{2}AIC_{\text{winning}}}}{e^{-\frac{1}{2}AIC_{\text{winning}}} + e^{-\frac{1}{2}AIC_{\text{cnst}}}} \tag{1.1}$$

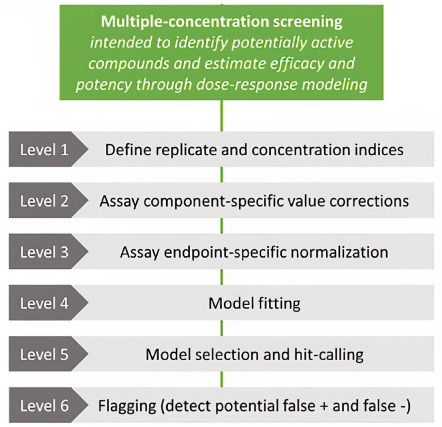
Finally, after processing, each CRS is categorized into an appropriate fit category based on the level of certainty in the estimated bioactivity. Additionally, cautionary flags are assigned to account for problematic data series or uncertain fits and hits.

1.2 New Toxicity Pipeline Implementation: pytcpl

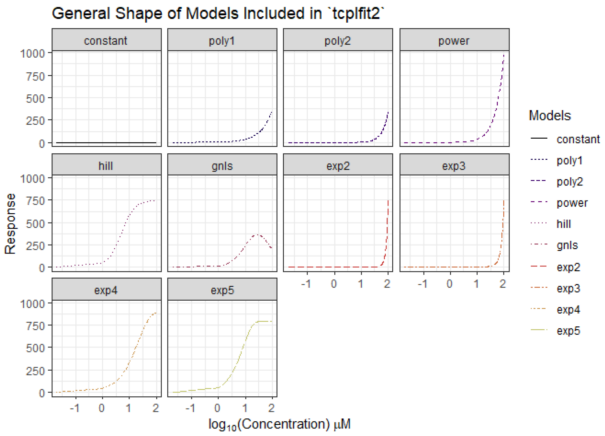
1.2.1 Introduction

This thesis introduces *pytcpl*[5], a streamlined `Python` package inspired by the `R` packages *tcpl* and tcplfit2. The package optimizes data storage and generates compressed `Parquet` files of the relevant raw data and metadata from *invitroDBv4.1*. This efficient strategy reduces storage needs, resulting in less than 10 GB within the repository, compared to the original 100 GB database. This avoids a cumbersome, large-scale database installation, rendering downstream analysis more accessible and efficient. Our package is crafted to accomodate cusomizable processing steps and facilitate interactive data visualization with an own *Curve Surfer*[6]. Furthermore, it enables

[5] https://github.com/rbBosshard/pytcpl
[6] https://pytcpl.streamlit.app/



(a) Overview of tcpl's data analysis pipeline for multiple concentrations screening data.



(b) Employed curve-fit models in tcpl v3.0 for fitting concentration-response data series through the application of maximum likelihood estimation.

Figure 1.3: tcpl

researchers who prefer `Python` to easily participate in data analysis and exploration, overcoming any limitations associated with using `R` code.

The pytcpl pipeline adds an additional setup and wrapup step to the main pipeline:

- **Setup:** Subset assay endpoints, add assay annotations from external data, enable workload balancing for distributed processing, generating `Parquet` files from all raw and metadata.
- **Main:** Cutoff determination, Curve fitting, Hit calling, Flagging.
- **Wrapup:** ICE curation, cytotoxicity interference reevaluation, exporting results and preparing for Curve Surfer.

1.2.2 Subset assay endpoints

For a better data comprehension, the presence matrix denoted as $P \in 0,1^{m \times n}$ is introduced. In this matrix, rows (indexed by i) represent assay endpoints a_i, and columns (indexed by j) indicate whether testing was performed[7] (1) or not performed (0) for compound c_j in those endpoints. Due to selective compound testing across different assay endpoints, matrix P is sparse. For a visual representation of the presence matrix P covering all assay endpoints and compounds in *invitroDBv4.1*, refer to Figure 1.4.

[7] A compound is regarded as having been tested within an assay endpoint when there exists a corresponding concentration-response series accessible in the database.

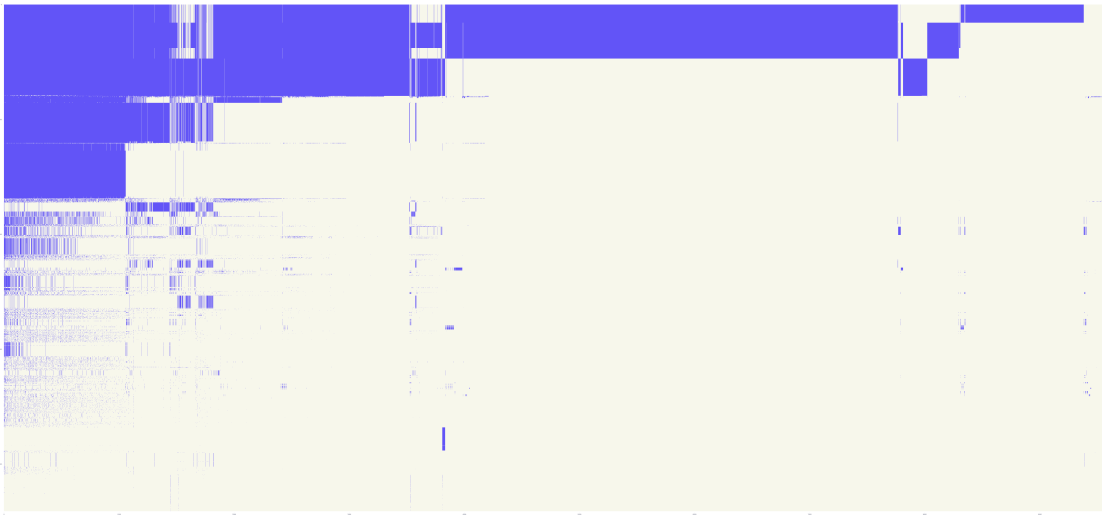


Figure 1.4: The *presence matrix* P covering all assay endpoints and compounds available in *invitroDBv3.5* with $m = 2205$ assay endpoints and $n = 9541$ compounds. The presence matrix is organized by sorting it based on the number of compounds present in each assay endpoint and the compounds are arranged in descending order of their presence frequency. The total count, where $P_{ij} = 1$, indicates the availability of 3 342 377 concentration-response series for downstream analysis.

For this thesis, only assay endpoints that have been tested with a minimum of 1000 compounds were exclusively taken into account. This criterion ensures the availability of sufficient data to train a machine learning model with a minimum level of robustness. Please consult Figure 1.5 to view a graphical depiction of the presence matrix P, which includes only the specific considerd subset of assay endpoints. From this point onward, this specific subset will be referred to as the *data* upon which the focus of this thesis will be directed.

In total, $\sum_{i,j} P_{ij} = 1\,372\,225$ concentration-response series are analyzed, encompassing a total of $\sum_{i,j} |S_{ij}| = 48\,861\,036$ concentration-response pairs across all compounds and assay endpoints.

Model	**Label**	**Equations**[1]	**Role in pytcpl**
Exponential 3	exp3	$f(x) = a\left(\exp\left(\left(\frac{x}{b}\right)^p\right) - 1\right)$	Omit
Gain-Loss 2	gnls2	$f(x) = \frac{tp}{1+\left(\frac{ga}{x}\right)^p} \exp\left(-qx\right)$	New

[1] Model parameters: a: x-scale, b: y-scale p: (gain) power, q: (loss) power, tp: top, ga: gain AC50

Table 1.3: tcplfit2 Model Details

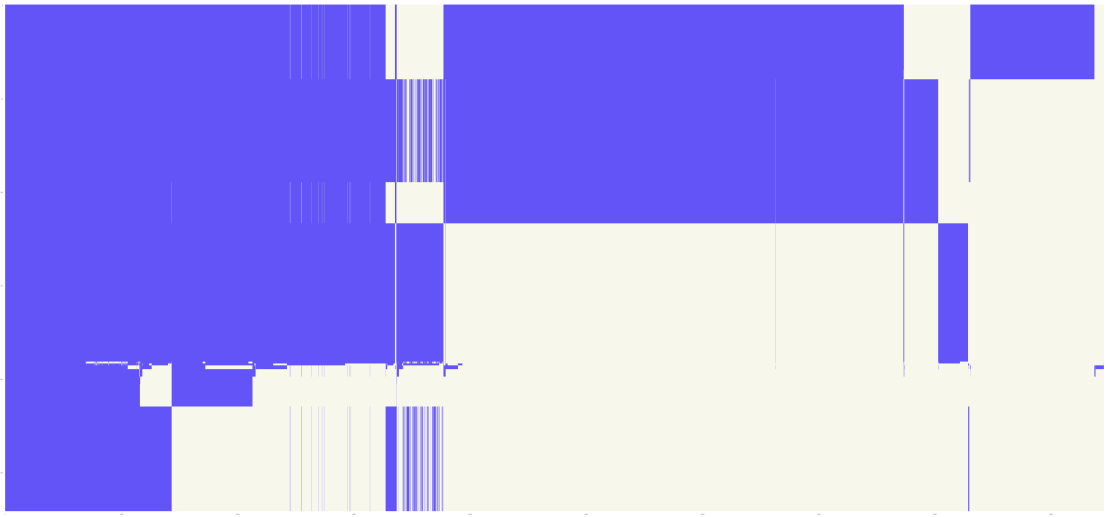


Figure 1.5: The *presence matrix* P covering only the subset of all of assay endpoints available in *invitroDBv3.5*, considered for this thesis, encompassing $m = 271$ assay endpoints and $n = 9456$ compounds. The total count, where $P_{ij} = 1$, indicates the availability of 1 372 225 concentration-response series for downstream analysis.

1.2.3 Cytotoxicity handling

We are determining cytotoxicity probability by assuming the ACC (or AC50, or other) has a Gaussian error distribution. We want to know the probability that:

$$ACC_{\text{target}} < ACC_{\text{cyto}}, \text{ which we'll call } P(\text{target action}).$$

We need:

ACC_{target}
ACC_{cyto}
$\text{SD}_{ACC_{\text{target}}}$ – we use 0.3 log units (this is in a few papers including the new ToxCast database one)
$\text{SD}_{ACC_{\text{cyto}}}$ – this is harder to define. We could base on the MAD but I think we just use 0.3 log units he

Then we find:

$$\begin{aligned} P(\text{target action}) &= P(ACC_{\text{target}} < ACC_{\text{cyto}}) \\ &= P(ACC_{\text{cyto}} - ACC_{\text{target}} > 0) \end{aligned}$$

For this we use the R function `pnorm`:

$$\texttt{pnorm}(ACC_{\text{cyto}} - ACC_{\text{target}}, 0, \sqrt{\text{SD}^2_{ACC_{\text{target}}} + \text{SD}^2_{ACC_{\text{cyto}}}})$$

This approach needs to be implemented differently in two cases:

A matching viability assay is available: Then we use the ACC_{cyto} from the matching burst assay.

A matching viability assay is not available, so we use the statistical approach. We use the median cytotox ACC for that chemical across all viability assays. Additionally, we account for *nhit* and *ntested*. We can do this by stating:

$$P(ACC_{\text{cyto}} < \text{max tested concentration}) = \frac{nhit}{ntested}$$

and then:

$$P(\text{cytotox action}) = P(ACC_{\text{target}} > ACC_{\text{cyto}}) \cdot P(ACC_{\text{cyto}} < \text{max tested concentration})$$

and so:

$$P(\text{target action}) = P(ACC_{\text{target}} < ACC_{\text{cyto}}) \cdot (1 - \frac{nhit}{ntested})$$

1.2.4 Curve Surfer

Data visualization, overview of what is possible with the tool. Filter by assay endpoint, compound, etc.

1.3 Machine Learning Pipeline

1.3.1 Preprocessing

Subselecting the columns from the output tables generated by pytcpl: DTXSID identifier and continuous hitcall value. The feature inputs to the machine learning model is a molecular structure represented as fingerprint generated from a SMILES string uniquely determined by the compounds DTXSID identifier. The SMILES string is a linear representation of a compound's molecular structure. The SMILES string is converted to a molecular graph, which is then converted to a feature vector. The feature vector is then used to train a machine learning model. The machine learning model is then used to predict the hitcall value for a given compound. The machine learning pipeline is illustrated in Figure?.

1.3.2 Binary Classification

The goal is to predict whether a compound is active or inactive for a given assay endpoint. This can be formulated as a binary classification problem, where the input consists of the molecular structure fingerprint of the compound, and the output is the binarized hitcall value based on a specific decision threshold. The hitcall value is rendered to a binary variable, where 1 indicates that the compound is active and 0 indicates that the compound is inactive.

1.3.3 Regression

1.3.4 Massbank Validation

Bibliography

[1] T. Sheffield, J. Brown, S. Davidson, K. P. Friedman, and R. Judson, "tcplfit2: an R-language general purpose concentration–response modeling package," *Bioinformatics*, vol. 38, no. 4, pp. 1157–1158, Nov. 2021, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab779. eprint: https://academic.oup.com/bioinformatics/article-pdf/38/4/1157/50422999/btab779.pdf. [Online]. Available: https://doi.org/10.1093/bioinformatics/btab779.

[2] C. for Computational Toxicology and U. E. Exposure, *Tcpl v3.0 data processing*, R package vignette for the tcpl package v3.0, CRAN, 2023. [Online]. Available: https://cran.r-project.org/web/packages/tcpl/vignettes/Data_processing.html.

Appendix A

Appendix

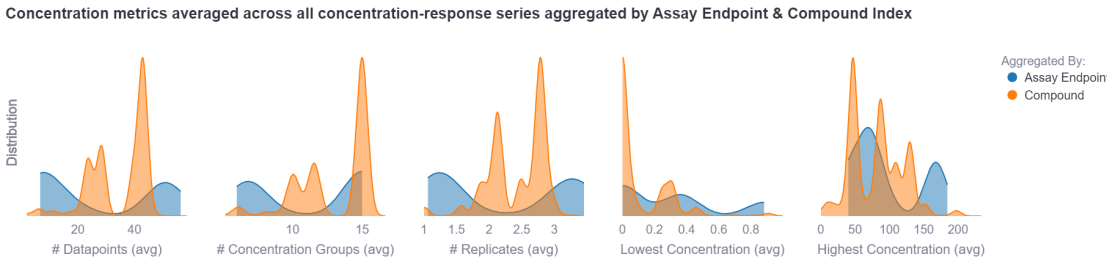


Figure A.1: Concentration metrics averaged across all concentration-response series aggregated by assay endpoint (blue) and compound (orange). E.g., the first chart shows the distribution (blue) on the average number of datapoints across all assay enpoint $a_i \in A$ with $\frac{1}{|C_i|} \sum_j n_{\text{datapoints}_{i,j}}$ and across all compounds $c_j \in C$ with $\frac{1}{|A_j|} \sum_i n_{\text{datapoints}_{i,j}}$. Similarly, the process is repeated for the other metrics: $n_{\text{groups}_{i,j}}$, $n_{\text{replicates}_{i,j}}$, $min_{\text{conc}_{i,j}}$, and $max_{\text{conc}_{i,j}}$.

Table A.1: Assay Source Names and Long Names

assay_source_name	assay_source_long_name
ACEA	ACEA Biosciences
APR	Apredica
ATG	Attagene
BSK	Bioseek
NVS	Novascreen
OT	Odyssey Thera
TOX21	Tox21/NCGC
CEETOX	Ceetox/OpAns
LTEA	LifeTech/Expression Analysis
VALA	VALA Sciences
CLD	CellzDirect
CCTE_PADILLA	CCTE Padilla Lab
TANGUAY	Tanguay Lab
STM	Stemina Biomarker Discovery
ARUNA	ArunA Biomedical
CCTE	CCTE Labs
CCTE_SHAFER	CCTE Shafer Lab
CPHEA_STOKER	CPHEA Stoker and Laws Labs
CCTE_GLTED	CCTE Great Lakes Toxicology and Ecology Division
UPITT	University of Pittsburgh Johnston Lab
UKN	University of Konstanz
ERF	Eurofins
TAMU	Texas A&M University
IUF	Leibniz Research Institute for Environmental Medicine
CCTE_MUNDY	CCTE Mundy Lab
UTOR	University of Toronto, Peng Laboratory